Expert Systems with Applications 37 (2010) 2081–2091

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Hierarchical decision rules mining

Qinrong Feng [a,b,c,*], Duoqian Miao [b,c], Yi Cheng [b,c]

[a] School of Mathematics and Computer Science, Shanxi Normal University, Linfen, Shanxi 041004, PR China
[b] Department of Computer Science and Technology, Tongji University, Shanghai 201804, PR China
[c] Key Laboratory of Embedded System & Service Computing, Ministry of Education of China, Tongji University, Shanghai, 201804, PR China

ARTICLE INFO

Keywords:
Multidimensional data model
Concept hierarchy
Hierarchical decision rules mining
Certain rules
Uncertain rules
Separate-and-conquer strategy

ABSTRACT

Decision rules mining is an important technique in machine learning and data mining. It has been studied intensively during the past few years. However, most existing algorithms are based on flat dataset, from which a set of decision rules mined may be very large for large scale data. Such a set of rules is not easily understandable and really useful for users. Moreover, too many rules may lead to over fitting. Thus, an approach to hierarchical decision rules mining is provided in this paper. It can mine decision rules from different levels of abstraction. The aim of this approach is to improve the quality and efficiency of decision rules mining by combining the hierarchical structure of multidimensional data model and the techniques of rough set theory. The approach follows the so-called separate-and-conquer strategy. It can not only provide a method of hierarchical decision rules mining, but also the most important is that it can reveal the fact that there exists property-preserving among decision rules mined from different levels, which can further improve the efficiency of decision rules mining.

1. Introduction

Decision rules mining is an important technique in data mining and machine learning. It has been used widely in business, medicine, finance, etc. A multitude of promising algorithms of decision rules mining (Fernández & Menasalvas, 2001; Hirokane, Konishi, Miyamoto, & Nishimura, 2007; Hong, Lin, Lin, & Wang, 2003; Hong, Lin, & Lin, 2008; Hu & Cercone, 1997, 2001) has been developed during the past few years. The aim of decision rules mining is to find a good rule set, which has a minimum number of rules and each rule should be as short as possible. We all know that each row of a decision table specifies a decision rule that determines decisions in terms of conditions. Thus, a set of decision rules mined from a decision table with large scale data may be very large. Such a set of rules are not easily understandable and really useful for users. Moreover, too many rules may lead to over fitting.

In many applications, we note that it is difficult to discover valuable patterns at low or primitive levels (raw data) due to the sparsity of data. Decision rules mined at high levels of abstraction may represent commonsense knowledge. Moreover, what may represent common sense to one user may seem novel to another. Therefore, it is necessary to have a framework of hierarchical decision rules mining to accommodate different user expectations or applications.

Some researchers have noticed that hierarchical attribute values exist impliedly in many real-world applications such as day, month, quarter and year for *Time* attribute. Hu and Cercone (1997, 2001) proposed a method to learn maximal-generalized decision rules from databases by integrating discretization, generalization, and rough set feature selection. Shan, Hamilton, and Cercone (1995) presented a three-step GRG approach for learning maximally general decision rules from large relational databases, which includes information generalization, information reduction, and rule generation. However, all of these algorithms are still process data at single level although the hierarchical structure of data was considered in advance.

Hong et al. (2003) proposed an algorithm to deal with the problem of producing a set of maximally general rules for coverage of training examples with hierarchical attribute values using rough sets. Hong et al. (2008) noticed that hierarchical attribute values are sually predefined in real-world applications and can be represented by hierarchical trees, so they provided a method to derive cross-level rules. Ziarko (2003) proposed a technique to create a linearly structured hierarchy of decision tables through hierarchical methods of boundary area reduction. But these algorithms do not have the ability to switch among different levels of abstraction flexibility.

Riquelme, Aguilar, and Toro (2000) provided a new approach HIDER (hierarchical decision rules) for learning rules in continuous and discrete domains based on evolution algorithm. The algorithm can produced a set of hierarchical rules. However, the limitation of this algorithm is that the rules must be applied in a specific order. Tsumoto (2002, 2003) examined closely the characteristics of

* Corresponding author. Address: School of Mathematics and Computer Science, Shanxi Normal University, Linfen, Shanxi 041004, PR China. Tel.: +86 0357 2051509; fax: +86 0357 2051187.
E-mail address: fengqr72@163.com (Q. Feng).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.06.065

medical experts' rules and provided a new approach to extract plausible rules, which consists of the following three procedures. First, the characterization of decision attributes is extracted from databases, and the classes are classified into several groups with respect to the characterization. Then, two kinds of subrules, characterization rules for each group and discrimination rules for each class in the group are induced. Finally, those two parts are integrated into one rule for each decision attribute. However, this algorithm extracts hierarchical decision rules by joining the similar rules, not by using the hierarchical attribute values.

With these observations, we note that multidimensional data models provide a way to organize data at different levels of granularity, which will be an effective data structure for processing hierarchical data. Thus, an approach to mine hierarchical decision rules is provided in this paper, it can switch among different granularities' flexibility by combining the structure of multidimensional data model (Pedersen & Jensen, 2001; Singh, Singh, & Suman, 2007) and the techniques of rough set theory (Duoqian & Daoguo, 2008; Pawlak, 1991; Wang, 2001; Zhang, Wu, & Liang, 2003).

Multidimensional data model is a variation of the relational model that uses multidimensional structures to organize data and express the relationships among data, in which the data are presented as a data cube, which is a lattice of cuboids. A multidimensional data model includes a number of dimensions that each includes multiple levels of abstraction defined by concept hierarchies (Lu, 1997), where hierarchy is a way to organize data at different levels of granularity. This organization provides users with the flexibility to view data from different perspectives. The best advantage of multidimensional data model is that data can be visualized and manipulated by a compact and easy-to-understand way. Based on the hierarchical structure of the multidimensional data model, it is possible to "scan" the given dataset at different levels of abstraction. At each level, the techniques of rough sets can then be used to discover and analyze significant patterns or rules.

Our algorithm follows the so-called separate-and-conquer strategy (Furnkranz, 1999), which can improve the efficiency of decision rules mining significantly. First, data are loaded in a data cuboid, and then it is generalized along dimensions until it gets to the most abstract level; thus, we can obtain data cuboid at the different abstract levels. This can decrease the number of tuples greatly. Secondly, data are processed by the top-down strategy and decision rules are mined by the technique of rough set theory. Then, the relationship of rules mined from different levels are analyzed, and we find that there exists property-preserving among decision rules mined from different levels of abstraction, that is, rules mined from different levels follow the false-preserving principle, which can further improve the quality and efficiency of hierarchical decision rules mining. Finally, an algorithm of hierarchical rules mining is designed, and we illustrate the procedure of this algorithm through an example. This algorithm can not only mine rules from any level of granularities to satisfy the need of users, but also improve the quality and efficiency of decision rules mining greatly.

The rest of the paper is organized as follows. Some preliminaries about rough set theory and multidimensional data model are reviewed in Section 2. In Section 3, the technique of data generalization based on multidimensional data model is introduced. In Section 4, the relation of decision rules mined from different levels of abstraction is analyzed, then an algorithm of hierarchical decision rules mining is proposed, and an example is given to illustrate the algorithm of hierarchical decision rules mining. Finally, the conclusion is given and the future works are pointed out.

2. Preliminaries

In this section, we will review some notions of rough set theory and multidimensional data model.

2.1. Rough set theory

Rough set theory was introduced by Pawlak as a mathematical tool to deal with imprecision, uncertainty or vague knowledge in artificial intelligence applications. It was used to discover data dependencies, data reduction, data mining, and rule extraction from databases, etc. Rough set theory is based on the ability to classify the observed and measured data, and some methods based on rough set theory are applied to deal with decision.

In this section, we will first review some basic notions of rough set theory, which can also be referred to Pawlak (1991, 2005) and Pawlak and Skowron (2007).

2.1.1. Indiscernibility relation

The starting point of rough set theory is the indiscernibility relation, which is generated by information about objects of interest. The indiscernibility relation expresses the fact that due to lack of information (or knowledge), we are unable to discern some objects by employing available information. This means that, in general, we are unable to deal with each particular object, but we have to consider granules (clusters) of indiscernible objects as a fundamental basis in rough set theory.

An information system is a quadruple $IS = (U, A, V, f)$, where U is a non-empty finite set of objects, called the universe, and A is a non-empty finite set of attributes, $V = \cup_{a \in A} V_a$, where V_a is the value set of a, called the domain of a, f is an information function from U to V, which assigns particular values from domains of attributes to objects such that $f_a(x_i) \in V_a$ for all $x_i \in U$ and $a \in A$.

Let $IS = (U, A, V, f)$ be an information system. For every subset of attributes $B \subseteq A$ and $B \neq \emptyset$, then $\cap B$ (intersection of all equivalence relations belong to B) that is also an equivalence relation, and will be denoted by $IND(B)$, which is defined by

$$xIND(B)y \text{ if and only if } a(x) = a(y), \text{ for every } a \in B,$$

where $a(x)$ denotes the value of attribute a for object x.

The family of all equivalence classes of $IND(B)$, i.e., the partition determined by B, will be denoted as $U/IND(B)$, or simply U/B, $[x]_B$ denotes the block of the partition U/B containing x; moreover, $[x]_{IND(B)} = \cap_{R \in B}[x]_R$.

2.1.2. Lower and upper approximations

The indiscernibility relation will be further used to define basic concepts of rough set theory. Suppose we are given an information system $IS = (U, A, V, f)$, with each subset $X \subseteq U$ and an equivalence relation $R \in A$, we associate two subsets:

$$\underline{R}X = \cup\{x \in U | [x]_R \subseteq X\}, \quad \overline{R}X = \cup\{x \in U | [x]_R \cap X \neq \emptyset\}$$

called the R-lower and R-upper approximate of X respectively.

Assuming P and Q are equivalence relations over U, then P-positive region of Q is the set of all objects of the universe U which can properly be classified to classes of U/Q employing knowledge expressed by the classification of U/P. It is defined by

$$POS_P(Q) = \bigcup_{X \in Q} \underline{P}X.$$

A positive region contains all patterns in U that can be classified in attribute set Q using the information in attribute set P.

2.1.3. Attribute reduction

A special case of information system called decision table, which is an information system of the form $DT = (U, C \cup D, V, f)$, where D is a distinguished attribute called the decision, and the elements of C are called conditions.

Given a decision table $DT = (U, C \cup D, V, f)$, for any $x, y \in U, a \in C$, if $a(x) = a(y)$, then $d(x) = d(y)$, we will call this decision table is consistent, otherwise, it is inconsistent.

In a decision table $DT = (U, C \cup D, V, f)$, there often exist some condition attributes that do not provide any additional information about the objects in U. So we should remove these attributes to reduce the complexity and cost of decision process.

Definition 1. Given a decision table $DT = (U, C \cup D, V, f)$, $r \in C$, if $POS_{C-\{r\}}(D) = POS_C(D)$, then r is a D-dispensable attribute in C, otherwise, r is a D-indispensable attribute in C.

Definition 2. Given a decision table $DT = (U, C \cup D, V, f)$ and an attribute set $P \subseteq C$, if for $\forall r \in P, POS_{P-\{r\}}(D) = POS_P(D)$, then P is independent with respect to D.

Definition 3. Given a decision table $DT = (U, C \cup D, V, f)$, an attribute set $P \subseteq C$ is a D-reduct of C if

(1) $POS_{C-\{r\}}(D) = POS_C(D)$;
(2) for $\forall r \in P$, $POS_{P-\{r\}}(D) \neq POS_P(D)$.

The intersection of all D-reducts is called a D-core (core with respect to D). Because the core is the intersection of all reducts, it is included in every reduct. Thus, in a sense, the core is the most important subset of attributes, since none of its elements can be removed without affecting the classification ability of attributes.

2.1.4. Decision rules

Each object of a decision table determines a decision rule. A decision rule $\phi \rightarrow \psi$ read "if ϕ then ψ", where ϕ is called the antecedent of the rule, and ψ is called the consequent of the rule.

Decision rules corresponding to some objects which have the same condition part but different decision part, such rules are called uncertain rules (inconsistent, nondeterministic, conflicting). Otherwise, the rules are called certain (consistent, sure, deterministic) rules.

Several numerical quantities of interest can be associated with a decision rule.

Definition 4. Given a decision table $DT = (U, C \cup D, V, f)$, the number $sup_x(\phi, \psi) = |\phi(x) \cap \psi(x)|$ will be called the support of the decision rule $\phi \rightarrow_x \psi$, which is the number of objects satisfying formula.

In some cases, it is not enough to know only the support of a rule. What we want to know mostly is which elements support the rule. So a supporting set is defined for a rule as follows.

Definition 5. Given a decision table $DT = (U, C \cup D, V, f)$, $\phi \rightarrow \psi$ is a rule in DT, we will say that an object $x \in U$ supports rule $\phi \rightarrow \psi$ iff x satisfies both ϕ and ψ, we call the set consists of x satisfies both ϕ and ψ the supporting set of $\phi \rightarrow \psi$ and denote it as $SS(\phi \rightarrow \psi)$.

A decision rule $\phi \rightarrow \psi$ may only reveal a part of the overall picture of the decision system from which it was derived. It may happen that the decision system contains objects that match the rule's antecedent ϕ but that have a different value for the decision attribute than the one indicated by the rule's consequent ψ. Hence, we are interested in the probability of the consequent ψ being correct given ϕ.

Definition 6. The certainty factor of a decision rule $\phi \rightarrow \psi$ is defined as $cer(\phi \rightarrow \psi) = |\phi(x) \cap \psi(x)|/|\phi(x)|$.

Obviously, if $cer(\phi \rightarrow \psi) = 1$, then the decision rule is certain, otherwise, it is uncertain.

2.2. Multidimensional data model

Multidimensional data model (Pedersen & Jensen, 2001; Singh et al., 2007) is a variation of the relational model that uses multidimensional structures to organize data and express the relationships among data, in which the data are presented as a data cube. A data cube, also known as a lattice of cuboid, defined by dimensions (or dimension attributes) and numeric facts called measures, allows data to be modeled and viewed from multiple dimensions. A multidimensional data model includes a number of dimension attributes that each includes multiple levels of abstraction defined by concept hierarchies, where hierarchy is a way to organize data at different levels of granularity. The elements of a dimension are called dimension value (or attribute value). Each such value belongs to a particular level. Measures in a data cube are usually a numerical function, which was computed by the operation of aggregate. It "lived" in cells defined uniquely by combinations of dimension values from each of the dimension in a data cube.

Data cube created for varying levels of abstraction are often referred to as cuboids, it is a lattice of cuboids. The cube created at the lowest level of abstraction is referred to as the base cuboid. A cube at the highest level of abstraction is the apex cuboid. For the purpose of our discussion, we will always use the term data cube to refer to a lattice of cuboids rather than an individual cuboid.

This organization provides users with the flexibility to view data from different granularities. Based on the hierarchical structure of multidimensional data model, it is possible to "scan" a data table from different levels of abstraction. At each level, the techniques of rough sets can then be used to discover and analyze significant patterns or rules. There exist a number of OLAP (on-line analytical processing) data cube operators to materialize these different views such as the operation of roll-up, drill-down, and slice and dice, etc. The operations of roll-up and drill-down are inverses of each other and make use of concept hierarchies and measures to perform aggregations. They can also make us handle data at various levels of abstraction flexibly. The operations of slice and dice can make us handle data from different angles, and the structure of a data cube provides us an intuitive way to generalize (or specialize) dimension values along one or more dimension attributes.

In a multidimensional data model, dimensions are hierarchical by nature. For example, dimension *Time* can be described by the dimension values: 'Year', 'Quarter', 'Month', and 'Day'. Alternatively, the dimension values of a dimension may be organized into a lattice, which indicates a partial order for the dimension. That is, the same *Time* dimension can have 'Year', 'Quarter', 'Month', 'Week', and 'Day' instead. With this scheme, the *Time* dimension is no longer a hierarchy because some weeks in the year may belong to different months.

In existing literature, measures in a data cube are mostly numerical data, which can also be other kinds of data such as spatial, multimedia, or text data (Han & Kamber, 2006). However, measures in our multidimensional data model are subsets of objects (an equivalence class induced uniquely by a combination of dimension values from each of the dimension attributes in the data cube). Consequently, the operation of aggregate in this multidimensional data model is reduced to the union of some subsets. That is, the measure in a cell at higher abstract level is equal to the union of its child cells at lower abstract level. So this representation can reduce the data scale greatly by the operation of roll-up along one or more dimensions. To our knowledge, this kind of multidimensional data model has not been reported in existing literature.

In a multidimensional data model, we can choose one dimension as a decision dimension according to the need for problems solving, and the others are conditional dimensions. Every dimension is composed of multiple levels, so we can classify the task of hierarchical decision rules mining based on data cube into three cases. (1) Keeping the decision dimension level unchanged and making every conditional dimension level changeable. (2) Keeping

all conditional dimension level unchanged and making decision dimension level changeable. (3) Conditional dimension levels and decision dimension level are all changeable. In this paper, we only focus on the first case, and other cases will be studied in the future.

3. Data generalization

A flat data table usually be given in real-world applications, from which the inherent hierarchical characteristics of data cannot been reflected effectively. While multidimensional data model can not only easily reflect the inherent hierarchical characteristics of data, but also provide a more compact representation for data. Moreover, we can obtain a compressed, high-level, generalized data by the hierarchical structure of multidimensional data model. So it is necessary to generalize the given flat data table to a multidimensional structure, that is, to transform a flat data table to a multidimensional data cube.

3.1. Concept hierarchy

We note that there exist some corresponding relations between a decision table and a multidimensional data cube. Each attribute in a decision table corresponds to a dimension attribute in a multidimensional data cube. Each concept hierarchy of an attribute in a decision table corresponds to a concept hierarchy of a dimension attribute in a multidimensional data cube. Each attribute value corresponds to a dimension value. Each combination of attribute values induces an equivalence class in a decision table, which can also be induced uniquely by a combination of dimension values from each of the dimensions in a multidimensional data cube, and this equivalence class induced from a combination of dimension values is placed in a cell of a multidimensional data cube. The universe of a decision table corresponds to the measure of a multidimensional data cube. Thus, a flat decision table corresponds to a data cuboid in a multidimensional data cube.

Having understood the corresponding relation between a decision table and a multidimensional data cube, we will only discuss concept hierarchy in multidimensional data model, which is also suits for attributes in a decision table.

In a multidimensional data model, data are organized into multiple dimensions, and each dimension contains multiple dimension values which form multiple abstract levels defined by concept hierarchies. A concept hierarchy defines a certain generalization relationships for dimension values in one or a set of dimensions. When a flat data table was generalized along one or more dimensions, and as a result, we will transform a large, low-level, detailed data into a compressed, high-level, generalized data.

Concept hierarchies are used to express knowledge in concise and high-level terms, which is usually in the form of tree. It can also facilitate mining knowledge at multiple levels of abstraction. In this paper, we assume that all concept hierarchies are simply balanced tree. Concept hierarchy is an important tool for capturing the generalization relations among objects and has been used in many data mining applications such as multilevel association rules mining, data warehouse, etc. As far as hierarchical decision rules mining is concerned, there is usually more than one possible way to build concept hierarchies for every dimension attribute because of different users may have different preferences. So concept hierarchies are usually built by combining the given dataset with some relevant domain knowledge, or it can be given by domain experts directly.

Concept hierarchies may also be defined by discretizing or grouping values for a given dimension attribute, that is, discretizing numerical data into interval and grouping categorical dimension values into a generalized abstract concept. A total or partial order can be defined among groups of values.

There also have some other methods to build concept hierarchies. Kuo and Huang (2005) and Kuo et al. (2006) provided a method of building concept hierarchies by modifying the traditional hierarchical clustering algorithms.

In what follows, we will illustrate concept hierarchies through an example.

Example 1. Build concept hierarchies for every attribute in Table 1.

Obviously, Table 1 is a flat data table, where the attribute "Salary" can be treated as a decision attribute, and attributes "Education-level" and "Vocation" are conditional attributes. So, every attribute in Table 1 corresponds to a dimension in a multidimensional data model, where the conditional attributes "Education-level" and "Vocation" correspond to conditional dimensions, and the decision attribute "Salary" corresponds to decision dimension. Obviously "Education-level" and "Vocation" are categorical attributes, and "Salary" is a numerical attribute. To build concept hierarchies for these dimensions, we should use the technique of discretizing for dimension "Salary" and grouping for dimensions "Education-level" and "Vocation", respectively, according to relevant domain knowledge.

We can build concept hierarchies structured as a tree for every dimension attribute. In every concept hierarchy tree, the root is the name of the dimension, leaves nodes are dimension values appearing in the given dataset, and internal nodes represent generalized dimension values of their child nodes.

As we all know, every attribute value determine an equivalence class in the given data table, so a subset of objects can be attached to every leaf node, it is an equivalence class determined by the attribute value (dimension value). For every internal node in a concept hierarchy tree, the subset of objects attached to it is equal to the union of subsets attached to its child nodes, and it is also an equivalence class. In essence, each node of a concept hierarchy is a concept, the dimension value or a node's label is its intent, and the subset of objects attached to it is its extent.

In a concept hierarchy, each level can be labeled by a number k. We stipulate that the level label of leaves nodes be assigned the number zero, and the level label for each internal node is one plus its child's level label. In this case, the concepts at each level can be called a concept at level k (see Fig. 1).

According to the relevant domain knowledge, we will build a two-level concept hierarchical structure for dimensions "Education-level" and "Vocation" as follows. Of course, not all of dimension hierarchies are necessarily to have the same depths in general. For simplicity of notation, we denote dimension "Education-level" as A, "Vocation" as B, and "Salary" as C, respectively. Every node in a concept hierarchy is denoted by the symbol in bracket near to it.

Concept hierarchies of all attributes in Table 1 are presented as Fig. 1. With these concept hierarchies tree, we can transform a flat data table to a multidimensional data cube.

Table 1
Training dataset.

U	Education-level	Vocation	Salary (unit: yuan)
1	Doctoral student	Private enterprise	Above 10000
2	Postgraduate student	State-owned enterprise	6000–10000
3	Others	Education	Under 2000
4	Undergraduate	Private enterprise	2000–6000
5	Undergraduate	State-owned enterprise	2000–6000
6	Postgraduate student	State-owned enterprise	2000–6000
7	Undergraduate	State-owned enterprise	6000–10000
8	Undergraduate	Civil servant	2000–6000
9	Doctoral student	Education	2000–6000
10	Others	State-owned enterprise	2000–6000

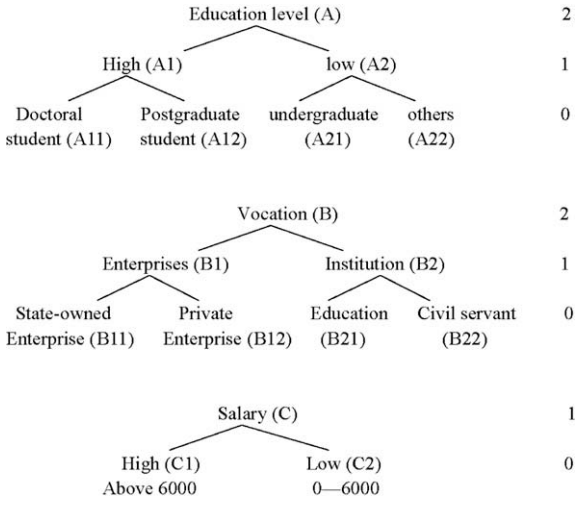

Fig. 1. Concept hierarchies.

3.2. Data transformation

Data transformation, also known as data consolidation, it is a phase in which the selected data are transformed into forms appropriate for the mining procedure.

In reality, dataset often has characteristics of multidimension and multilevel. However, the dataset we obtained is usually represented as a flat data table, which is difficult to reflect hierarchical characteristics of data. In general, decision rules mined from different level will have different value, and different users may also have different preferences. Thus, we will transform a flat data table to a multidimensional data cube to mine more valuable information from the given dataset.

Multidimensional data model provides a tool to enhance data with generalization along one or more dimensions. Generalizing or enhancing data (Zhang, Zhang, & Yang, 2004) is critical in generating a dataset that is cleaner, clearer, and smaller than the original, which can improve the efficiency of decision rules mining significantly. So, we should enhance information from raw data to discover more valuable and high quality rules. The raw data in a database is called at its primitive level, and the knowledge is said to be at a primitive level if it is discovered by using raw data only.

In addition, multidimensional data model also have some other advantages. For example, it can provide multiple perspectives to view data from multidimension and multilevel. A major benefit of multidimensional data model is that it can provide a compact and easy to understand way to visualize and manipulate data elements that have many interrelationships. The operations of roll-up along concept hierarchies in a multidimensional data model can reduce the data horizontally. Horizontal reduction is accomplished by merging identical tuples after the substitution of a dimension value by its higher level value in a pre-defined concept hierarchy for categorical dimensions, or the discretization of numerical dimensions. This can decrease the number of tuples greatly.

Multidimensional data model has widely been used in OLAP during the past few years, which mainly tend to trend analysis. Thus, measures in a data cube are usually numerical data. However, in this paper, we will mainly focus on the task of hierarchical decision rules mining, the advantage of our multidimensional data model is that measures in our data cube are subsets of objects, which is an equivalence class induced uniquely by the combination of dimension values from each of the dimensions in a data cube. So, the operation of aggregate is reduced to union of subsets during the course of roll-up. With rolling up along concept hierarchies level by level, measures "lived" in cells will be merged step by step until getting to the top most abstract level.

In what follows, we will illustrate the procedure of data transformation through an example.

Example 2. According to concept hierarchies illustrated as Fig. 1, transform the Table 1 to a multidimensional data cube.

We will transform Table 1 to a data cube after having understood the corresponding relation between a relational table and a multidimensional data cube. In this example, we will transform Table 1 to a data cube by rolling-up along hierarchies of all conditional dimensions simultaneously. The result of transformation according to level zero for every conditional dimension is illustrated as Fig. 2.

We will further roll-up data cuboid illustrated as Fig. 2 along every dimension attribute simultaneously, and we will obtain a multidimensional data cuboid as Fig. 3 according to level 1 for every conditional dimension.

The relational table corresponds to the multidimensional data cuboid illustrated as Fig. 3 is Table 2. Because every dimension attribute has rolled up to its top most abstract level, and thus the data cuboid illustrated in Fig. 3 cannot be rolled up further. So, we obtain a two-level multidimensional data model. How do we mine hierarchical decision rules based on this multidimensional data model? The solution can be found in the following section.

4. Hierarchical decision rules mining

In this section, an approach to hierarchical decision rules mining is provided, which can mine rules from different levels of

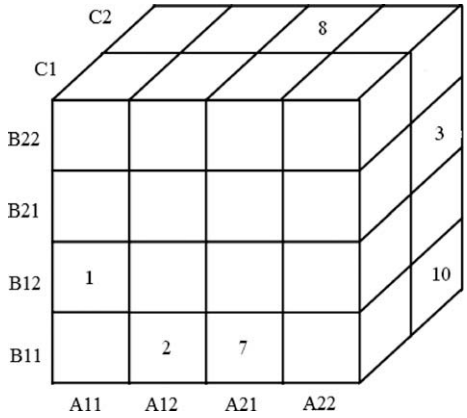

Fig. 2. Data cuboid corresponding concept hierarchies 0.

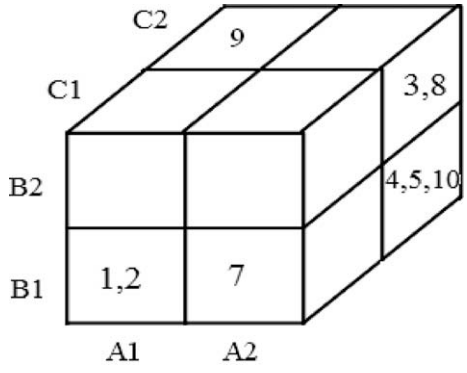

Fig. 3. Data cuboid corresponding concept hierarchies 1.

Table 2
Data table corresponding to Fig. 3.

U	Education-level	Vocation	Salary
{1,2}	High	Enterprise	High
{3,8}	Low	Institution	Low
{4,5,10}	Low	Enterprise	Low
{6}	High	Enterprise	Low
{7}	Low	Enterprise	High
{9}	High	Institution	Low

abstraction. At each level, the techniques of rough sets can then be used to discover significant patterns or rules.

The aim of decision rules mining is to find a concise rule set, which has a minimum number of rules and each rule should be as short as possible. To this end, we should first simplify the given decision table, i.e., eliminate the irrelevant or superfluous attributes and attribute values without losing essential information about the original decision table. As a result, a set of concise and meaningful rules will be produced. Then, we will mine decision rules at different levels based on the hierarchical structure of multidimensional data model.

After the phase of data generalization mentioned above, we adopt the top-down strategy to mine hierarchical decision rules based on the multidimensional data model. The procedure of it consists of the following two steps:

(1) Data reduction: attribute reduction and attribute value reduction.
(2) Decision rules generation and analysis: generate decision rules using rough sets and analyze the relationship among decision rules mined from different levels of abstraction.

4.1. Data reduction

4.1.1. Attribute reduction

Attribute reduction is one of the most important concepts in rough set theory. It has intensively been studied during the past few years. As we all known, a decision table usually has more than one reduct, which will induce different sets of rules. In practice, it is always hoped to obtain the set of the most concise rules. Therefore, people have been attempting to find a minimal reduct of a decision table, which means that the number of attributes contained in the reduct is minimal. Unfortunately, it has been proved that finding a minimal reduct of a decision table is an NP-hard problem (Wong & Ziarko, 1985). Thus, we have to seek a near-optimal solution for attribute reduction. Heuristic reduction algorithms are a good choice if time cost is considered.

To design an effective heuristic algorithm, the most important thing is to effectively measure and then rank the relative significance of different attributes in a decision system. The attribute significance can be defined from different aspects. To the best of our knowledge, existing heuristic reduction algorithms can be classified into three classes: among the first class of algorithms, the attribute significance is defined based on positive region (Pawlak, 1991), the second class of algorithms is based on discernibility matrix (Duoqian, 1997; Jue & Duoqian, 1998; Skowron & Rauszer, 1991); and the last class of algorithms is based on information entropy (Duoqian & Guirong, 1999; Duoqian & Jue, 1997).

Of course, many attribute reduction algorithms have been provided not limited the above-mentioned things. So, in this paper, we will not discuss algorithms of attribute reduction in detail.

4.1.2. Attribute value reduction

The process which the maximum number of condition attribute values are removed without loosing essential information is called attribute value reduction, and the resulting rule is called minimal length. Computing minimal length rules is of particular importance in knowledge discovery since they represent general patterns existing in decision table.

Attribute reduction can only discard some redundant attributes in decision table, but it cannot discard all of the redundant information sufficiently. To this end, we can obtain a more simplified decision table by further processing it. That is, we should discard those redundant attribute values by performing value reduction.

When data are rolled up to the most abstract level in a multidimensional model, the number of tuples will be decreased considerably, in that many objects with same attribute values will be merged into one. This can reduce the complexity of attribute value reduction greatly. For the sake of simplicity, we assume that the set of condition attributes is already reduced, i.e. there have no superfluous condition attributes in the decision table. There are some algorithms for attribute value reduction in (Duoqian & Daoguo, 2008; Pawlak, 1991; Wang, 2001). So, we will not describe them here.

4.2. Hierarchical decision rules analysis

Decision rules are generated easily after the procedure of data reduction mentioned above. Each row of a decision table specifies a decision rule that determines decisions in terms of conditions. So, it will generate too many rules for large scale data.

Multidimensional data model provides an effective tool to reduce the amount of data. In a multidimensional data model, the amount of data can be reduced greatly by the operation of roll-up along one or more dimensions. In other words, more and more non-empty cells in cuboids will be merged during the course of rolling up, which will reduce the number of non-empty cells greatly.

Given a decision table $DT = (U, C \cup D, V, f)$ and all concept hierarchies of its conditional attributes, we might call it the $(0,0,\ldots,0)$th decision table as well, that is, all of conditional attribute values are at leaves level of their concept hierarchies respectively. The $(0,0,\ldots,0)$th decision table can also be denoted as

$$DT_{\underbrace{00\cdots0}_{m}} = (U_{\underbrace{00\cdots0}_{m}}, C \cup D, V^{\underbrace{00\cdots0}_{m}}, f_{\underbrace{00\cdots0}_{m}}),$$

where m is the number of conditional attributes, $U_{\underbrace{00\cdots0}_{m}}$ is the universe, $C = \{C_1, C_2, \ldots, C_m\}$ is the set of conditional attributes and D is the decision attribute, $V^{\underbrace{00\cdots0}_{m}}$ is the domain of conditional attributes, and $f_{\underbrace{00\cdots0}_{m}}$ is the information function from $U_{\underbrace{00\cdots0}_{m}}$ to $V^{\underbrace{00\cdots0}_{m}}$. We will denote the depth of concept hierarchy of C_1 as $l(C_1)$, the depth of concept hierarchy of C_2 as $l(C_2)$, and the rest may be deduced by analogy. Without loss of generality, we denote a generalized decision table as

$$DT_{k_1k_2\cdots k_m} = \left(U_{k_1k_2\cdots k_m}, C \cup D, V^{k_1k_2\cdots k_m}, f_{k_1k_2\cdots k_m}\right)$$

and call it the $(k_1,k_2,\ldots,k_m)$th decision table, where $C = \{C_1, C_2, \ldots, C_m\}$ is the set of conditional attributes and D is the decision attribute, $U_{k_1k_2\cdots k_m}$ is the universe of the $(k_1,k_2,\ldots,k_m)$th decision table, $V^{k_1k_2\cdots k_m} = \cup_{t=1}^{m} V_t^{k_t}$ is the domain of the $(k_1,k_2,\ldots,k_m)$th decision table, where $V_t^{k_t}$ is the domain of C_t at its kth level of its concept hierarchy, $f_{k_1k_2\cdots k_m}$ is the information function from $U_{k_1k_2\cdots k_m}$ to $V^{k_1k_2\cdots k_m}$.

For a decision table, when some of its attribute values are generalized to a higher level of abstraction, correspondingly, its

universe will be reduced, its information function will be changed, and thus, the decision table will also be changed. Thus, attributes values with different degrees of abstraction will determine different decision tables, or we can say that a decision table is determined uniquely by the degree of abstraction of attributes values. So we will denote a decision table with different degrees of abstraction only by the sequence of level labels of every conditional attributes, e.g., we can denote the $(k_1, k_2, \ldots, k_m)$th decision table

$$DT_{k_1k_2\cdots k_m} = (U_{k_1k_2\cdots k_m}, C \cup D, V^{k_1k_2\cdots k_m}, f_{k_1k_2\cdots k_m})$$

only by

$$(k_1, k_2, \ldots, k_m),$$

which means that the combination of the domain of C_1 at k_1th level of its concept hierarchy and the domain of C_2 at k_2th level of its concept hierarchy, and the rest may be deduced by analogy.

We note that a new data cuboid will be produced once the roll-up operation is performed, and the produced data cuboid can also be transformed to a decision table. That is, a data cuboid at every abstract level corresponds to a decision table; thus, we will call the data cuboid corresponds to the $(k_1, k_2, \ldots, k_m)$th decision table the $(k_1, k_2, \ldots, k_m)$th data cuboid. Or we can use the term 'the $(k_1, k_2, \ldots, k_m)$th decision table' and 'the $(k_1, k_2, \ldots, k_m)$th data cuboid' exchangeably in what follows.

In order to analyze the relation of rules mined from data cuboid with different degrees of abstraction, we will first give some definitions as follows.

Denote the $(k_1, k_2, \ldots, k_m)$th data cuboid by $DT_{k_1k_2\cdots k_m} = (U_{k_1k_2\cdots k_m}, C \cup D, V^{k_1k_2\cdots k_m}, f_{k_1k_2\cdots k_m})$, where $U_{k_1k_2\cdots k_m}$ is the measure of the $(k_1, k_2, \ldots, k_m)$th data cuboid, C is conditional dimensions, D is the decision dimension, $V^{k_1k_2\cdots k_m}$ is the domain of all conditional dimensions at the $(k_1, k_2, \ldots, k_m)$th level, and $f_{k_1k_2\cdots k_m}$ is the information function from $U_{k_1k_2\cdots k_m}$ to $V^{k_1k_2\cdots k_m}$.

Definition 7. (Wille, 1992) A concept of the context (G, M, I) is a pair (A, B) with $A \subseteq G, B \subseteq M$, and $A' = B, B' = A$, where

$$A' = \{m \in M |\ gIm \text{ for all } g \in A\},$$
$$B' = \{g \in G |\ gIm \text{ for all } m \in B\},$$

then we call (A_1, B_1) is a subconcept of (A_2, B_2) if and only if $A_1 \subseteq A_2$ (or $B_1 \supseteq B_2$) and denote it as $(A_1, B_1) \leqslant (A_2, B_2)$.

Definition 8. Given a decision table $DT = (U, C \cup D, V, f)$, where $C = \{C_1, C_2, \ldots, C_m\}$ is the set of conditional dimension attributes, for any $C_t \in C$, denote the domain of C_t at the ith level in its concept hierarchy as V_t^i, we will say that V_t^i is finer than V_t^j if and only if for any $a \in V_t^i$, there always exist $b \in V_t^j$ such that a is a subconcept of b, and denote it as $V_t^i \leqslant V_t^j$.

Obviously, if $i \leqslant j$, then we have $V_t^i \leqslant V_t^j$. That is, when the domain of C_i is rolled up along its concept hierarchy from a lower level to a higher level, the values of C_i is generalized.

Definition 9. Given a decision table $DT = (U, C \cup D, V, f)$, where $C = \{C_1, C_2, \ldots, C_m\}$ is the set of conditional dimension attributes, denote the domain of C at $(i_1, i_2, \ldots, i_m)$th data cuboid as $V^{i_1i_2\cdots i_m}$, we will say that $V^{i_1i_2\cdots i_m}$ is finer than $V^{j_1j_2\cdots j_m}$ if and only if for any $k \in \{1, 2, \ldots, m\}$, there always have $V_k^{i_k} \leqslant V_k^{j_k}$, and we will denote it as $V^{i_1i_2\cdots i_m} \leqslant V^{j_1j_2\cdots j_m}$.

Definition 10. Given the $(i_1, i_2, \ldots, i_m)$th data cuboid

$$DT_{i_1i_2\cdots i_m} = (U_{i_1i_2\cdots i_m}, C \cup D, V^{i_1i_2\cdots i_m}, f_{i_1i_2\cdots i_m})$$

and the $(j_1, j_2, \ldots, j_m)$th data cuboid

$$DT_{j_1j_2\cdots j_m} = (U_{j_1j_2\cdots j_m}, C \cup D, V^{j_1j_2\cdots j_m}, f_{j_1j_2\cdots j_m}),$$

we will say that the measure $U_{i_1i_2\cdots i_m}$ is finer than $U_{j_1j_2\cdots j_m}$ if and only if for every $X \in U_{i_1i_2\cdots i_m}$, there always exists $Y \in U_{j_1j_2\cdots j_m}$ such that $X \subseteq Y$, we denote it as $U_{i_1i_2\cdots i_m} \leqslant U_{j_1j_2\cdots j_m}$.

Definition 11. Given the $(i_1, i_2, \ldots, i_m)$th data cuboid

$$DT_{i_1i_2\cdots i_m} = (U_{i_1i_2\cdots i_m}, C \cup D, V^{i_1i_2\cdots i_m}, f_{i_1i_2\cdots i_m})$$

and the $(j_1, j_2, \ldots, j_m)$th data cuboid

$$DT_{j_1j_2\cdots j_m} = (U_{j_1j_2\cdots j_m}, C \cup D, V^{j_1j_2\cdots j_m}, f_{j_1j_2\cdots j_m}),$$

if $V^{i_1i_2\cdots i_m} \leqslant V^{j_1j_2\cdots j_m}$, then we will call data cuboid $DT_{i_1i_2\cdots i_m}$ is finer than $DT_{j_1j_2\cdots j_m}$, and denote it as $DT_{i_1i_2\cdots i_m} \leqslant DT_{j_1j_2\cdots j_m}$. If there exists at least one dimension $C_t \in C$ such that $V_t^{i_t} < V_t^{j_t}$, then we will say that data cuboid $DT_{i_1i_2\cdots i_m}$ is strictly finer than $DT_{j_1j_2\cdots j_m}$ and denote it as $DT_{i_1i_2\cdots i_m} < DT_{j_1j_2\cdots j_m}$.

Of course, the coarser/finer relation of data cuboids with different degrees of abstraction is determined only by the granularity of its dimension values. Consequently, the granularity of the measure and corresponding information function can be induced from the granularity of dimension values.

Property 1. *Given the $(i_1, i_2, \ldots, i_m)$th data cuboid*

$$DT_{i_1i_2\cdots i_m} = (U_{i_1i_2\cdots i_m}, C \cup D, V^{i_1i_2\cdots i_m}, f_{i_1i_2\cdots i_m})$$

and the $(j_1, j_2, \ldots, j_m)$th data cuboid

$$DT_{j_1j_2\cdots j_m} = (U_{j_1j_2\cdots j_m}, C \cup D, V^{j_1j_2\cdots j_m}, f_{j_1j_2\cdots j_m})$$

if $i_1 \leqslant j_1, i_2 \leqslant j_2, \ldots, i_m \leqslant j_m$, then we have $V^{i_1i_2\cdots i_m} \leqslant V^{j_1j_2\cdots j_m}$.

This property shows that the domain of C will become more abstract when the domain of C is rolled up from a lower level $(i_1, i_2, \ldots, i_m)$ to a higher level $(j_1, j_2, \ldots, j_m)$.

Property 2. *Given the $(i_1, i_2, \ldots, i_m)$th data cuboid*

$$DT_{i_1i_2\cdots i_m} = (U_{i_1i_2\cdots i_m}, C \cup D, V^{i_1i_2\cdots i_m}, f_{i_1i_2\cdots i_m})$$

and the $(j_1, j_2, \ldots, j_m)$th data cuboid

$$DT_{j_1j_2\cdots j_m} = (U_{j_1j_2\cdots j_m}, C \cup D, V^{j_1j_2\cdots j_m}, f_{j_1j_2\cdots j_m})$$

if $i_1 \leqslant j_1, i_2 \leqslant j_2, \ldots, i_m \leqslant j_m$, then we have $U_{i_1i_2\cdots i_m} \leqslant U_{j_1j_2\cdots j_m}$.

This property shows that the lower the level of a data cuboid, the finer the measure of this data cuboid. Or we can say that non-empty cells in data cube will be merged when a data cuboid is rolled up from a lower level of abstraction to a higher level of abstraction.

Property 3. *Given the $(i_1, i_2, \ldots, i_m)$th data cuboid*

$$DT_{i_1i_2\cdots i_m} = (U_{i_1i_2\cdots i_m}, C \cup D, V^{i_1i_2\cdots i_m}, f_{i_1i_2\cdots i_m})$$

and the $(j_1, j_2, \ldots, j_m)$th data cuboid

$$DT_{j_1j_2\cdots j_m} = (U_{j_1j_2\cdots j_m}, C \cup D, V^{j_1j_2\cdots j_m}, f_{j_1j_2\cdots j_m})$$

if $i_1 \leqslant j_1, i_2 \leqslant j_2, \ldots, i_m \leqslant j_m$, then we have $DT_{i_1i_2\cdots i_m} \leqslant DT_{j_1j_2\cdots j_m}$.

This property indicates that the lower the level of a data cuboid, the finer the data cuboid.

Property 4. *Given a decision table $DT = (U, C \cup D, V, f)$, where $C = \{C_1, C_2, \ldots, C_m\}$ is the set of conditional dimension attributes, denote the set of data cuboids with different degree of abstraction as*

$$DTS = \{DT_{k_1k_2\cdots k_m} | 0 \leqslant k_1 \leqslant l(C_1) - 1, 0 \leqslant k_2 \leqslant l(C_2) - 1, \ldots, 0 \leqslant k_m \leqslant l(C_m) - 1\},$$

where $l(C_t)$ denote the depth of concept hierarchy of dimension C_t. Then, $(DTS, \leqslant)$ can be organized as a lattice, where the relation $\leqslant$ is the coarser/finer relation among data cuboids.

Proof. Obviously, the relation $\leqslant$ is a partial order relation over *DTS*, so $(DTS, \leqslant)$ is a poset. In order to prove the poset $(DTS, \leqslant)$ can be organized as a lattice, we only need to prove that any two elements of *DTS* have the supremum and infimum.

For any two data cuboids $DT_{i_1 i_2 \cdots i_m}, DT_{j_1 j_2 \cdots j_m} \in DTS$, denote $DT_{i_1 i_2 \cdots i_m} \vee DT_{j_1 j_2 \cdots j_m}$ as the supremum of $DT_{i_1 i_2 \cdots i_m}$ and $DT_{j_1 j_2 \cdots j_m}$, $DT_{i_1 i_2 \cdots i_m} \wedge DT_{j_1 j_2 \cdots j_m}$ as the infimum of $DT_{i_1 i_2 \cdots i_m}$ and $DT_{j_1 j_2 \cdots j_m}$. If we set

$$(k_1 k_2 \cdots k_m) = (\max\{i_1, j_1\} \max\{i_2, j_2\} \cdots \max\{i_m, j_m\}),$$
$$(l_1 l_2 \cdots l_m) = (\min\{i_1, j_1\} \min\{i_2, j_2\} \cdots \min\{i_m, j_m\}).$$

Then, we have

$$DT_{i_1 i_2 \cdots i_m} \vee DT_{j_1 j_2 \cdots j_m} = DT_{k_1 k_2 \cdots k_m}$$

and

$$DT_{i_1 i_2 \cdots i_m} \wedge DT_{j_1 j_2 \cdots j_m} = DT_{l_1 l_2 \cdots l_m},$$

that is, $DT_{k_1 k_2 \cdots k_m}$ is the supremum of $DT_{i_1 i_2 \cdots i_m}$ and $DT_{j_1 j_2 \cdots j_m}$, $DT_{l_1 l_2 \cdots l_m}$ is the infimum of $DT_{i_1 i_2 \cdots i_m}$ and $DT_{j_1 j_2 \cdots j_m}$. Thus, $(DTS, \leqslant)$ is a lattice. □

Property 5. *Given the $(i_1, i_2, \ldots, i_m)$th data cuboid*

$$DT_{i_1 i_2 \cdots i_m} = (U_{i_1 i_2 \cdots i_m}, C \cup D, V^{i_1 i_2 \cdots i_m}, f_{i_1 i_2 \cdots i_m})$$

and denote its positive region as $POS_{i_1 i_2 \cdots i_m}$, if $i_1 \leqslant j_1, i_2 \leqslant j_2, \ldots, i_m \leqslant j_m$, then we have $POS_{j_1 j_2 \cdots j_m} \subseteq POS_{i_1 i_2 \cdots i_m}$. More generally, we have

$$POS_{(l(C_1)-1)\cdots(l(C_m)-1)} \subseteq \cdots \subseteq POS_{\underbrace{00\ldots0}_{m}},$$

where $l(C_t)$ denote the depth of concept hierarchy of dimension C_t.

This property indicates that the positive region is decrease monotonously with the data cuboid drilled down from higher level to lower level.

Property 6. *Given the $(i_1, i_2, \ldots, i_m)$th data cuboid*

$$DT_{i_1 i_2 \cdots i_m} = (U_{i_1 i_2 \cdots i_m}, C \cup D, V^{i_1 i_2 \cdots i_m}, f_{i_1 i_2 \cdots i_m}),$$

a rule induced from this data cuboid is certain if and only if the supporting set of the rule is included in the positive region of this data cuboid.

This result can be obtained easily by combining the definition of certain rule and positive region of a data cuboid.

Property 7. *Given the $(i_1, i_2, \ldots, i_m)$th data cuboid*

$$DT_{i_1 i_2 \cdots i_m} = (U_{i_1 i_2 \cdots i_m}, C \cup D, V^{i_1 i_2 \cdots i_m}, f_{i_1 i_2 \cdots i_m})$$

and the $(j_1, j_2, \ldots, j_m)$th data cuboid

$$DT_{j_1 j_2 \cdots j_m} = (U_{j_1 j_2 \cdots j_m}, C \cup D, V^{j_1 j_2 \cdots j_m}, f_{j_1 j_2 \cdots j_m}),$$

where $DT_{i_1 i_2 \cdots i_m} \leqslant DT_{j_1 j_2 \cdots j_m}$, then certain decision rules mined from $DT_{j_1 j_2 \cdots j_m}$ remain certain in $DT_{i_1 i_2 \cdots i_m}$.

Proof. A rule is certain if and only if it is induced from the lower approximate of a concept. Obviously, if a rule is induced from the positive region of a data cuboid, it must be a certain one. In other words, if the supporting set of a rule is included in the positive region of the data cuboid, it is certain.

Since $DT_{i_1 i_2 \cdots i_m} \leqslant DT_{j_1 j_2 \cdots j_m}$, namely, data cuboid $DT_{i_1 i_2 \cdots i_m}$ is finer than $DT_{j_1 j_2 \cdots j_m}$, so we have $POS_{j_1 j_2 \cdots j_m} \subseteq POS_{i_1 i_2 \cdots i_m}$. According to Property 6, we have that a rule is certain in $DT_{j_1 j_2 \cdots j_m}$ means that the supporting set of this rule is included in $POS_{j_1 j_2 \cdots j_m}$. So it is included in $POS_{i_1 i_2 \cdots i_m}$. Thus, the proposition holds. □

From Property 7, we find that there exists property preserving among decision rules mined from different levels of abstraction, which can be summarized as false-preserving principle (Ling & Bo, 2007). This can further improve the efficiency of decision rules mining.

Property 8 (False-preserving principle). *Given a base data cuboid and its generalization at different levels of abstraction. If uncertain rules cannot been mined from a coarser-grained data cuboid, then they also cannot been mined from the base data cuboid. Especially, if uncertain rules cannot been mined from the top most abstract level, then we can say that the base data cuboid is consistent.*

4.3. Algorithm and examples

Based on the above discussion, a decision table can be constructed based on the apex data cuboid, in which, the number of objects will decrease greatly because many duplicative objects with the same attribute values are merged. Thus, the number of decision rules is also decreased greatly than that at the primitive level. We call the decision table corresponds to the apex data cuboid the coarsest decision table.

In the coarsest decision table, each row of it determines a decision rule. Of course, redundant information maybe exists in this decision table. So, we should further simplify it. Attribute reduction will eliminate redundant columns without losing the classification ability. This can simplify the decision table vertically and horizontally since some duplicative tuples will be merged. Attribute value reduction will discard redundant attribute values on the premise of keeping the classification ability unchanged. This can make those rules mined from the simplified decision table much fewer and shorter than the original decision table.

As it has been discussed in the previous section, rough set theory is particularly well suited to deal with inconsistency in a decision table. If a decision table is inconsistent, then lower and upper approximations of the decision classes are computed. For each decision class, certain rules are generated from objects belonging to its lower approximation. Uncertain rules are generated either from the upper approximation or from the boundaries of this decision class. We also notice that certain rules indicate unique decision value while uncertain rules lead to a few possible decisions.

As Han (1995) noticed that: *With the mining of rules at multiple concept levels, similar rules could be generated at different concept levels. Some of these rules can be considered as redundant and be eliminated from the knowledge base to avoid the inclusion of many superfluous rules. In principle, a rule is considered redundant if it does not convey additional information and is less general than another rule.* Thus, to eliminate redundant decision rules effectively, our algorithm of hierarchical decision rules mining follows the so-called separate-and-conquer strategy (Furnkranz, 1999), which has been coined by Pagallo and Haussler (1990), because of the way of developing a theory that characterizes this learning strategy: learn a rule that covers a part of the given training examples, remove the covered examples from the training set (the separate part) and recursively learn another rule that covers some of the remaining examples(the conquer part) until no examples remain.

In our algorithm, we first load data into a pre-assigned multidimensional data cube, roll-up along dimensions to the most abstract level, and build the coarsest decision table from it; then mine certain rules from this coarsest decision table, and remove the supporting sets of certain rules from the universe; finally, drill down along one or more dimensions to a lower level and recursively mine other rules that support the remaining examples until no examples remain or getting to the primitive level. Thus, the output of our algorithm will include rules mined from different

abstract levels. This can obtain not only generalized rules, but also rules mined from various levels of abstraction.

For a fixed decision level, we adopt the top-down strategy starting from the coarsest level of conditional dimensions to the finest level. We have proved that certain rules mined from higher level remain certain at lower level. So, we can save these certain rules into a rule set and remove their supporting set from the current universe; then drill down to a lower level and further process those uncertain rules. The algorithm will terminate until no objects remain or getting to the primitive level.

The aim of this algorithm is to find a set of certain decision rules which are concise to the greatest extent, that is, the number of rules is as fewer as possible and the length of every rule is as shorter as possible. Furthermore, this algorithm provides a way to mine all certain rules with different degree of generalization from different abstract levels.

To summarize, the procedure of hierarchical decision rules mining from data cube is described as the following algorithm.

Algorithm (*Hierarchical decision rules mining*).

Input: A given decision table $DT = (U, C \cup D, V, f)$.
Output: A set of decision rules RS.
Step 1: Build concept hierarchies for every attribute in DT, choose a level in concept hierarchy of decision attribute, and fix it in the whole algorithm.
Step 2: Transform the given decision table to a multidimensional data cube by use of concept hierarchies constructed as above.
Step 3: Denote the decision table corresponds to the apex data cuboid as

$$DT_{l(C_1)l(C_2)\cdots l(C_m)} = \left(U_{l(C_1)l(C_2)\cdots l(C_m)}, C \cup D, V^{l(C_1)l(C_2)\cdots l(C_m)}, f_{l(C_1)l(C_2)\cdots l(C_m)}\right)$$

set $l(C_1) + l(C_2) + \cdots + l(C_m) = k$, $RS = \emptyset$, U is the universe of the given decision table, where $l(C_t)$ is the depth of concept hierarchy of C_t.
Step 4: Build the decision table DT_k based on the data cuboid at the *kth* level, set $RS_k = \emptyset$.
Step 5: Simplify the decision table DT_k (implement attribute reduction and attribute value reduction), generate decision rules, combine those duplicative decision rules as one, and merge their supporting set of the duplicative decision rules.
Step 6: For every decision rules, if the certainty factor of a rule is equal to 1, then it is a certain one. Otherwise, it is uncertain.Save those certain decision rules to RS_k and set $RS \cup RS_k$ to RS, that is, $RS := RS \cup RS_k$. Compute the supporting set $SS(RS_k)$ of RS_k, set $U - SS(RS_k)$ to U.
Step 7: If $U = \emptyset$ or $k = 0$, then turn to step 8, otherwise, $k := k - 1$ (this means drill-down one level along one dimension. In fact, we can drill-down multiple levels along more dimensions simultaneously or according to the need of users), then turn to step 4.
Step 8: Merge those rules induced from the remaining examples to RS and output it. The algorithm is terminated.

This algorithm performs the following three main tasks:

(1) Data generalization. This task is accomplished by steps 1 and 2, which mainly focus on data transformation, that is, transform the given flat decision table to a multidimensional data cube, which provide a structure to process data at different levels of abstraction.
(2) Data reduction. This task is completed by steps 4 and 5, which can simplify the decision table at each level of abstraction. It can make rules mined from this decision table as shorter and fewer as possible.
(3) Mine certain rules and remove their supporting set from the current universe at every level of abstraction.

Eventually, the algorithm returns a hierarchical certain rule set with various degrees of abstraction and some uncertain rules at the primitive level.

An example will be illustrated to demonstrate how the algorithm works. The decision table is shown in Table 3, which contain 10 objects and two conditional attributes. We notice that there exists inconsistencies tuples in this dataset.

Example 3. Suppose a two-layer decision system is considered as in Fig. 1. The training data used in this example is as Table 3. Illustrate the procedure of algorithm by data Table 3.

(1) Build concept hierarchies for every attribute in Table 3 as illustrated in Fig. 1, and then transforming the Table 3 to a data cuboid as Fig. 2.

Roll-up the data cuboid as illustrated in Fig. 2 along dimensions "Educational-level" and "Vacation", we will obtain the data cuboid as Fig. 3.
For the data cuboid in Fig. 3, its corresponding decision table is as Table 4.
After merged the duplicative objects of Table 4, we will obtain a simplified table as Table 2.

(2) Compute the reduct of Table 2 or Table 4, a minimal reduct of Table 2 or Table 4 is {Vocation}. Of course the core of it is also {Vocation}.

Now, we have climbed to the top most abstract level due to its two-level structure for this multidimensional data cube, and we will mine decision rules begin from this most abstract level.

Table 3
Training dataset.

U	Education-level	Vocation	Salary
1	Doctoral student	Private enterprise	High
2	Postgraduate student	State-owned enterprise	High
3	Others	Education	Low
4	Undergraduate	Private enterprise	Low
5	Undergraduate	State-owned enterprise	Low
6	Postgraduate student	State-owned enterprise	Low
7	Undergraduate	State-owned enterprise	High
8	Undergraduate	Civil servant	Low
9	Doctoral student	Education	Low
10	Others	State-owned enterprise	Low

Table 4
Dataset roll-up to concept level 1.

U	Education-level	Vocation	Salary
1	High	Enterprise	High
2	High	Enterprise	High
3	Low	Institution	Low
4	Low	Enterprise	Low
5	Low	Enterprise	Low
6	High	Enterprise	Low
7	Low	Enterprise	High
8	Low	Institution	Low
9	High	Institution	Low
10	Low	Enterprise	Low

Table 5
Rules mined from Table 2 or Table 4.

	Decision rules	Supporting set	Certainty factor
1	If Vocation = "Enterprise" then Salary = "High"	{1,2,7}	0.5
2	If Vocation = "Enterprise" then Salary = "Low"	{4,5,6,10}	0.5
3	If Vocation = "Institution" then Salary = "Low"	{3,8,9}	1

Table 6
Remaining dataset.

U	Education-level	Vocation	Salary
1	Doctoral student	Private enterprise	High
2	Postgraduate student	State-owned enterprise	High
4	Undergraduate	Private enterprise	Low
5	Undergraduate	State-owned enterprise	Low
6	Postgraduate student	State-owned enterprise	Low
7	Undergraduate	State-owned enterprise	High
10	Others	State-owned enterprise	Low

(3) Rules mined from Table 2 are presented in Table 5. Rules mined from the top most abstract level are presented in Table 5. We can see that there are only three items of decision rules, which are fewer than the number of objects in the raw dataset. These generalized rules will provide us more concise and easy-to-understand information than the original dataset. Of course, if you are unsatisfied with this result, you can proceed to this procedure by drilling down to a lower level.

From Table 5, we know that the rule "If Vocation = 'Institution' then Salary = 'Low'" is a certain one, so its supporting set {3,8,9} can be removed from the universe according to the algorithm, and those uncertain rules are processed further by drilling-down along all conditional dimensions to the next level, which is the primitive level in this example.

(4) In this situation, the Table 3 is translated into the Table 6, a minimal reduct of Table 6 is {Education-level, Vocation}, and the core is also {Education-level, Vocation}.

(5) Rules mined from Table 6 are as follows.

Certain rules

(1) If Educational-level = "Doctoral student" then Salary = "High"
(2) If Educational-level = "Undergraduate" AND Vocation = "Private enterprise" then Salary = "Low"
(3) If Educational-level = "Others" then Salary = "Low"

Uncertain rules

(4) If Educational-level = "Postgraduate student" AND Vocation = "State-owned enterprise" then Salary = "High"
(5) If Educational-level = "Postgraduate student" AND Vocation = "State-owned enterprise" then Salary = "LOW"
(6) If Educational-level = "Undergraduate" AND Vocation = "State-owned enterprise" then Salary = "LOW"
(7) If Educational-level = "Undergraduate" AND Vocation = "State-owned enterprise" then Salary = "High"

(6) So, the final output is as follows.

Certain rules

(1) If Vocation = "Institution" then Salary = "Low"
(2) If Educational-level = "Doctoral student" then Salary = "High"
(3) If Educational-level = "Undergraduate" AND Vocation = "Private enterprise" then Salary = "Low"
(4) If Educational-level = "Others" then Salary = "Low"

Uncertain rules

(5) If Educational-level = "Postgraduate student" AND Vocation = "State-owned enterprise" then Salary = "High"
(6) If Educational-level = "Postgraduate student" AND Vocation = "State-owned enterprise" then Salary = "LOW"
(7) If Educational-level = "Undergraduate" AND Vocation = "State-owned enterprise" then Salary = "LOW"
(8) If Educational-level = "Undergraduate" AND Vocation = "State-owned enterprise" then Salary = "High"

We can see that the number of decision rules mined from data cuboid at higher abstract level is much fewer than those mined from lower level. In this example, there are only three items of decision rules mined from the most abstract level, while there are 10 items of decision rules mined from primitive level. Furthermore, we can mine all certain rules with different degrees of generalization from different abstract levels.

5. Conclusion and discussion

Decision rules mining is an important technique in machine learning and data mining. It has been studied intensively during the past few years. However, most existing algorithms are based on flat data tables. Set of decision rules mined from flat data table can be very large for large scale data. Such set of rules are not easy to be understood.

In real-world applications, we notice that data are hierarchical in nature. So, a method of hierarchical decision rules mining is provided in this paper, which aims to improve the quality and efficiency of decision rules mining by combining the hierarchical structure of multidimensional data model and the techniques of rough set theory. This algorithm can not only output some generalized rules with different degree of generalization, but also reveal the fact of property-preserving among decision rules mined from different levels of abstraction.

Moreover, our algorithm will obtain different results for different type of decision tables, i.e. consistent decision table and inconsistent tables. For a consistent decision table, the universe will be reduced to an empty set when the algorithm gets to the finest level. That is, we can mine all certain rules with different degree of generalization at different levels. While for an inconsistent decision table, the universe can not be an empty set when the algorithm gets to the primitive level.

6. Future trends

Hierarchical structure of data has received much attention in recent years. It has been used in many areas such as hierarchical reduction, hierarchical text classification, etc. However, there has not an effective representation and operations for hierarchical data at present.

Multidimensional data model is a variation of the relational model that uses multidimensional structures to organize data, and the main advantage of it is that it offers a powerful way to represent hierarchical structured data. Moreover, there are a number of typical data cube operations, which are very fit to hierarchical data.

Thus, based on the hierarchical structure of multidimensional data model, there are some works need to be studied further in

the forthcoming future, such as attribute reduction at multiple levels, attribute value reduction at multiple levels, and hierarchical approximate decision rules mining, etc. These will be paid close attention by more and more researchers.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Serial No. 60475019, 60775036) and The Research Fund for the Doctoral Program of Higher Education (Serial No. 20060247039).

References

Duoqian, M. (1997). *Rough set theory and its application in machine learning*. PhD dissertation, Institute of Automation, Chinese Academy of Sciences (in Chinese).

Duoqian, M., & Daoguo, L. (2008). *Rough sets theory, algorithms and applications*. Beijing, China: Tsinghua University Press.

Duoqian, M., & Guirong, H. (1999). A heuristic algorithm for reduction of knowledge. *Journal of Computer Research and Development, 36*(6), 681–684.

Duoqian, M., & Jue, W. (1997). Information-based algorithm for reduction of knowledge. In *Proceedings of 1997 IEEE international conference on intelligent processing systems* (pp. 1155–1158) (Vol. 2).

Fernández, M. C., Menasalvas, F., et al. (2001). Minimal decision rules based on the APRIORI algorithm. *International Journal of Applied Mathematics and Computer Science, 11*(3), 691–704.

Furnkranz, J. (1999). Separate-and-conquer rule learning. *Artificial Intelligence Review, 13*, 3–54.

Han, J. (1995). Mining knowledge at multiple concept levels. In *Proceedings of the 1995 international conference on information and knowledge management* (pp. 19–24).

Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques* (2nd ed.). Morgan Kaufmann Publisher.

Hirokane, M., Konishi, H., Miyamoto, A., & Nishimura, F. (2007). Extraction of minimal decision algorithm using rough sets and genetic algorithm. *Systems and Computers in Japan, 38*(4), 39–51.

Hong, T.-P., Lin, C.-E, & Lin, J.-H., & Wang, S.L. (2003). Learning a coverage set of multiple-level certain and possible rules by rough sets. In *Proceedings of the IEEE international conference on systems, man and cybernetics*. (pp. 2605–2610) (Vol. 3).

Hong, T.-P., Lin, C.-E., & Lin, J.-H. (2008). Learning cross-level certain and possible rules by rough sets. *Expert Systems with Applications, 34*(3), 1698–1706.

Hu, X., & Cercone, N. (1997). Learning maximal generalized decision rules via discretization, generalization and rough set feature selection. In *Proceedings of the 9th IEEE international conference of tools on artificial intelligence, Newport Beach, California, USA* (pp. 548–556).

Hu, X., & Cercone, N. (2001). Discovering maximal generalized decision rules through horizontal and vertical data reduction. *Computational Intelligence, 17*(4), 685–702.

Jue, W., & Duoqian, M. (1998). Analysis on attribute reduction strategies of rough set. *Journal of Computer Science and Technology, 13*(2), 189–192.

Kuo, H.-C., & Huang, J.-P. (2005). Building a concept hierarchy from a distance matrix. In M. A. Klopotek et al. (Eds.), *Intelligent information processing and web mining* (pp. 87–95). Springer.

Kuo, H.-C., Tsai, T.-H., & Huang J.-P. (2006). Building a concept hierarchy by hierarchical clustering with join/merge decision. In H.D. Cheng et al. (Eds.), *Proceedings of the 9th joint conference on information sciences, Kaohsiung, Taiwan, ROC, October 8–11, 2006* (pp. 142–145). Atlantis Press.

Ling, Z., & Bo, Z. (2007). *Theory and applications of problem solving* (2nd ed.). Beijing: Tsinghua University Publisher.

Lu, Y. (1997). *Concept hierarchy in data mining: specification, generation and implementation*. Master thesis, Simon Fraser University, Canada.

Pagallo, G., & Haussler, D. (1990). Boolean feature discovery in empirical learning. *Machine Learning, 5*(1), 71–99.

Pawlak, Z. (1991). *Rough sets: Theoretical aspects of reasoning about data*. Dordrecht: Kluwer Academic Publishing.

Pawlak, Z. (2005). Rough sets and decision algorithms. In W. Ziarko and Y. Yao (Eds.), *RSCTC 2000, lecture notes of artificial intelligence 2005* (pp. 30–45).

Pawlak, Z., & Skowron, A. (2007). Rudiments of rough sets. *Information Sciences, 177*(1), 3–27.

Pedersen, T. B., & Jensen, C. S. (2001). Multidimensional database technology. *Computer, 34*(12), 40–46.

Riquelme, J. C., Aguilar, J. S., & Toro, M. (2000). Discovering hierarchical decision rules with evolutive algorithms in supervised learning. *The International Journal of Computers, Systems and Signal, 1*(1), 73–84.

Shan, N., Hamilton, H. J., & Cercone, N. (1995). GRG: knowledge discovery using information generalization, information reduction, and rule generation. In *Proceedings of the seventh international conference on tools with artificial intelligence* (pp. 372–379).

Singh, M., Singh, P., & Suman, T. (2007). Conceptual multidimensional model. In *Proceedings of world academy of science, engineering and technology* (pp. 709–714) (Vol. 26).

Skowron, A., & Rauszer, C. (1991). The discernibility matrices and functions in information systems. In R. Slowinski (Eds.), *Intelligent decision support-handbook of application and advances of the rough sets theory* (pp. 331–362).

Tsumoto, S. (2002). Mining hierarchical decision rules from clinical databases using rough sets and medical diagnostic model. In T. Elomaa et al. (Eds.), *PKDD, lecture notes on artificial intelligence* (pp. 423–435) (Vol. 2431).

Tsumoto, S. (2003). Automated extraction of hierarchical decision rules from clinical databases using rough set model. *Expert Systems with Applications, 24*(2), 189–197.

Wang, G. (2001). *Rough set theory and knowledge acquisition*. Xi'an, China: Xi'an Jiaotong University Press.

Wille, R. (1992). Concept lattices and conceptual knowledge systems. *Computers and Mathematics with Applications, 23*(6–9), 493–515.

Wong, S. K. M., & Ziarko, W. (1985). On optimal decision rules in decision tables. *Bulletin of Polish Academy of Sciences, 33*, 693–696.

Zhang, W. X., Wu, W. Z., & Liang, Jiye (2003). *Rough set Theory and Methods*. Beijing, China: Science Press.

Zhang, S., Zhang, C., & Yang, Q. (2004). Information enhancement for data mining. *IEEE Intelligent System, 19*(2), 12–13.

Ziarko, W. (2003). Acquisition of hierarchy-structured probabilistic decision tables and rules from data. *Expert Systems, 20*(5), 305–310.